

Package ‘paws.database’

March 9, 2021

Title Amazon Web Services Database Services

Version 0.1.11

Description Interface to Amazon Web Services database services, including 'Relational Database Service' ('RDS'), 'DynamoDB' 'NoSQL' database, and more <<https://aws.amazon.com/>>.

License Apache License (>= 2.0)

URL <https://github.com/paws-r/paws>

BugReports <https://github.com/paws-r/paws/issues>

Imports paws.common (>= 0.3.0)

Suggests testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Collate 'dax_service.R' 'dax_interfaces.R' 'dax_operations.R'
'docdb_service.R' 'docdb_interfaces.R' 'docdb_operations.R'
'dynamodb_service.R' 'dynamodb_interfaces.R'
'dynamodb_operations.R' 'dynamodbstreams_service.R'
'dynamodbstreams_interfaces.R' 'dynamodbstreams_operations.R'
'elasticsearch_service.R' 'elasticsearch_interfaces.R'
'elasticsearch_operations.R' 'neptune_service.R'
'neptune_interfaces.R' 'neptune_operations.R' 'rds_service.R'
'rds_operations.R' 'rds_custom.R' 'rds_interfaces.R'
'rdsdataservice_service.R' 'rdsdataservice_interfaces.R'
'rdsdataservice_operations.R' 'redshift_service.R'
'redshift_interfaces.R' 'redshift_operations.R'
'simpledb_service.R' 'simpledb_interfaces.R'
'simpledb_operations.R'

NeedsCompilation no

Author David Kretch [aut, cre],
Adam Banker [aut],
Amazon.com, Inc. [cph]

Maintainer David Kretch <david.kretch@gmail.com>

Repository CRAN

Date/Publication 2021-03-09 16:40:02 UTC

R topics documented:

dax	2
docdb	4
dynamodb	5
dynamodbstreams	8
elasticache	10
neptune	12
rds	15
rdsdataservice	19
redshift	20
simpledb	24
Index	26

dax	<i>Amazon DynamoDB Accelerator (DAX)</i>
-----	------------------------------------------

Description

DAX is a managed caching service engineered for Amazon DynamoDB. DAX dramatically speeds up database reads by caching frequently-accessed data from DynamoDB, so applications can access that data with sub-millisecond latency. You can create a DAX cluster easily, using the AWS Management Console. With a few simple modifications to your code, your application can begin taking advantage of the DAX cluster and realize significant improvements in read performance.

Usage

```
dax(config = list())
```

Arguments

config Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- dax(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
```

```

    ),
    profile = "string"
  ),
  endpoint = "string",
  region = "string"
)
)

```

Operations

create_cluster	Creates a DAX cluster
create_parameter_group	Creates a new parameter group
create_subnet_group	Creates a new subnet group
decrease_replication_factor	Removes one or more nodes from a DAX cluster
delete_cluster	Deletes a previously provisioned DAX cluster
delete_parameter_group	Deletes the specified parameter group
delete_subnet_group	Deletes a subnet group
describe_clusters	Returns information about all provisioned DAX clusters if no cluster identifier is specified, or a
describe_default_parameters	Returns the default system parameter information for the DAX caching software
describe_events	Returns events related to DAX clusters and parameter groups
describe_parameter_groups	Returns a list of parameter group descriptions
describe_parameters	Returns the detailed parameter list for a particular parameter group
describe_subnet_groups	Returns a list of subnet group descriptions
increase_replication_factor	Adds one or more nodes to a DAX cluster
list_tags	List all of the tags for a DAX cluster
reboot_node	Reboots a single node of a DAX cluster
tag_resource	Associates a set of tags with a DAX resource
untag_resource	Removes the association of tags from a DAX resource
update_cluster	Modifies the settings for a DAX cluster
update_parameter_group	Modifies the parameters of a parameter group
update_subnet_group	Modifies an existing subnet group

Examples

```

## Not run:
svc <- dax()
svc$create_cluster(
  Foo = 123
)

## End(Not run)

```

docdb

*Amazon DocumentDB with MongoDB compatibility***Description**

Amazon DocumentDB API documentation

Usage

```
docdb(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- docdb(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_tags_to_resource	Adds metadata tags to an Amazon DocumentDB resource
apply_pending_maintenance_action	Applies a pending maintenance action to a resource (for example, to an Amazon DocumentDB instance)
copy_db_cluster_parameter_group	Copies the specified cluster parameter group
copy_db_cluster_snapshot	Copies a snapshot of a cluster
create_db_cluster	Creates a new Amazon DocumentDB cluster
create_db_cluster_parameter_group	Creates a new cluster parameter group
create_db_cluster_snapshot	Creates a snapshot of a cluster
create_db_instance	Creates a new instance
create_db_subnet_group	Creates a new subnet group
delete_db_cluster	Deletes a previously provisioned cluster
delete_db_cluster_parameter_group	Deletes a specified cluster parameter group
delete_db_cluster_snapshot	Deletes a cluster snapshot
delete_db_instance	Deletes a previously provisioned instance

<code>delete_db_subnet_group</code>	Deletes a subnet group
<code>describe_certificates</code>	Returns a list of certificate authority (CA) certificates provided by Amazon DocumentDB
<code>describe_db_cluster_parameter_groups</code>	Returns a list of DBClusterParameterGroup descriptions
<code>describe_db_cluster_parameters</code>	Returns the detailed parameter list for a particular cluster parameter group
<code>describe_db_clusters</code>	Returns information about provisioned Amazon DocumentDB clusters
<code>describe_db_cluster_snapshot_attributes</code>	Returns a list of cluster snapshot attribute names and values for a manual DB cluster snapshot
<code>describe_db_cluster_snapshots</code>	Returns information about cluster snapshots
<code>describe_db_engine_versions</code>	Returns a list of the available engines
<code>describe_db_instances</code>	Returns information about provisioned Amazon DocumentDB instances
<code>describe_db_subnet_groups</code>	Returns a list of DBSubnetGroup descriptions
<code>describe_engine_default_cluster_parameters</code>	Returns the default engine and system parameter information for the cluster parameter group
<code>describe_event_categories</code>	Displays a list of categories for all event source types, or, if specified, for a specific event source type
<code>describe_events</code>	Returns events related to instances, security groups, snapshots, and DB parameter groups
<code>describe_orderable_db_instance_options</code>	Returns a list of orderable instance options for the specified engine
<code>describe_pending_maintenance_actions</code>	Returns a list of resources (for example, instances) that have at least one pending maintenance action
<code>failover_db_cluster</code>	Forces a failover for a cluster
<code>list_tags_for_resource</code>	Lists all tags on an Amazon DocumentDB resource
<code>modify_db_cluster</code>	Modifies a setting for an Amazon DocumentDB cluster
<code>modify_db_cluster_parameter_group</code>	Modifies the parameters of a cluster parameter group
<code>modify_db_cluster_snapshot_attribute</code>	Adds an attribute and values to, or removes an attribute and values from, a manual DB cluster snapshot
<code>modify_db_instance</code>	Modifies settings for an instance
<code>modify_db_subnet_group</code>	Modifies an existing subnet group
<code>reboot_db_instance</code>	You might need to reboot your instance, usually for maintenance reasons
<code>remove_tags_from_resource</code>	Removes metadata tags from an Amazon DocumentDB resource
<code>reset_db_cluster_parameter_group</code>	Modifies the parameters of a cluster parameter group to the default value
<code>restore_db_cluster_from_snapshot</code>	Creates a new cluster from a snapshot or cluster snapshot
<code>restore_db_cluster_to_point_in_time</code>	Restores a cluster to an arbitrary point in time
<code>start_db_cluster</code>	Restarts the stopped cluster that is specified by DBClusterIdentifier
<code>stop_db_cluster</code>	Stops the running cluster that is specified by DBClusterIdentifier

Examples

```
## Not run:
svc <- docdb()
svc$add_tags_to_resource(
  Foo = 123
)

## End(Not run)
```

Description

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation, and use the AWS Management Console to monitor resource utilization and performance metrics.

DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements, while maintaining consistent and fast performance. All of your data is stored on solid state disks (SSDs) and automatically replicated across multiple Availability Zones in an AWS region, providing built-in high availability and data durability.

Usage

```
dynamodb(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- dynamodb(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

[batch_execute_statement](#)
[batch_get_item](#)
[batch_write_item](#)
[create_backup](#)
[create_global_table](#)
[create_table](#)

This operation allows you to perform batch reads and writes on data stored in DynamoDB.
 The BatchGetItem operation returns the attributes of one or more items from one or more tables.
 The BatchWriteItem operation puts or deletes multiple items in one or more tables.
 Creates a backup for an existing table.
 Creates a global table from an existing table.
 The CreateTable operation adds a new table to your account.

delete_backup	Deletes an existing backup of a table
delete_item	Deletes a single item in a table by primary key
delete_table	The DeleteTable operation deletes a table and all of its items
describe_backup	Describes an existing backup of a table
describe_continuous_backups	Checks the status of continuous backups and point in time recovery on the specified table
describe_contributor_insights	Returns information about contributor insights, for a given table or global secondary index
describe_endpoints	Returns the regional endpoint information
describe_export	Describes an existing table export
describe_global_table	Returns information about the specified global table
describe_global_table_settings	Describes Region-specific settings for a global table
describe_kinesis_streaming_destination	Returns information about the status of Kinesis streaming
describe_limits	Returns the current provisioned-capacity quotas for your AWS account in a Region
describe_table	Returns information about the table, including the current status of the table, when it was created, and the current status of the table's replicas
describe_table_replica_auto_scaling	Describes auto scaling settings across replicas of the global table at once
describe_time_to_live	Gives a description of the Time to Live (TTL) status on the specified table
disable_kinesis_streaming_destination	Stops replication from the DynamoDB table to the Kinesis data stream
enable_kinesis_streaming_destination	Starts table data replication to the specified Kinesis data stream at a timestamp chosen by the user
execute_statement	This operation allows you to perform reads and singleton writes on data stored in DynamoDB
execute_transaction	This operation allows you to perform transactional reads or writes on data stored in DynamoDB
export_table_to_point_in_time	Exports table data to an S3 bucket
get_item	The GetItem operation returns a set of attributes for the item with the given primary key
list_backups	List backups associated with an AWS account
list_contributor_insights	Returns a list of ContributorInsightsSummary for a table and all its global secondary indexes
list_exports	Lists completed exports within the past 90 days
list_global_tables	Lists all global tables that have a replica in the specified Region
list_tables	Returns an array of table names associated with the current account and endpoint
list_tags_of_resource	List all tags on an Amazon DynamoDB resource
put_item	Creates a new item, or replaces an old item with a new item
query	The Query operation finds items based on primary key values
restore_table_from_backup	Creates a new table from an existing backup
restore_table_to_point_in_time	Restores the specified table to the specified point in time within EarliestRestorableTime
scan	The Scan operation returns one or more items and item attributes by accessing every item in the table
tag_resource	Associate a set of tags with an Amazon DynamoDB resource
transact_get_items	TransactGetItems is a synchronous operation that atomically retrieves multiple items
transact_write_items	TransactWriteItems is a synchronous write operation that groups up to 25 action requests
untag_resource	Removes the association of tags from an Amazon DynamoDB resource
update_continuous_backups	UpdateContinuousBackups enables or disables point in time recovery for the specified table
update_contributor_insights	Updates the status for contributor insights for a specific table or index
update_global_table	Adds or removes replicas in the specified global table
update_global_table_settings	Updates settings for a global table
update_item	Edits an existing item's attributes, or adds a new item to the table if it does not already exist
update_table	Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams
update_table_replica_auto_scaling	Updates auto scaling settings on your global tables at once
update_time_to_live	The UpdateTimeToLive method enables or disables Time to Live (TTL) for the specified table

Examples

```
## Not run:
svc <- dynamodb()
# This example reads multiple items from the Music table using a batch of
# three GetItem requests. Only the AlbumTitle attribute is returned.
svc$batch_get_item(
  RequestItems = list(
    Music = list(
      Keys = list(
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Call Me Today"
          )
        ),
        list(
          Artist = list(
            S = "Acme Band"
          ),
          SongTitle = list(
            S = "Happy Day"
          )
        ),
        list(
          Artist = list(
            S = "No One You Know"
          ),
          SongTitle = list(
            S = "Scared of My Shadow"
          )
        )
      ),
    ProjectionExpression = "AlbumTitle"
  )
)

## End(Not run)
```

Description

Amazon DynamoDB

Amazon DynamoDB Streams provides API actions for accessing streams and processing stream records. To learn more about application development with Streams, see [Capturing Table Activity with DynamoDB Streams](#) in the Amazon DynamoDB Developer Guide.

Usage

```
dynamodbstreams(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- dynamodbstreams(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

describe_stream	Returns information about a stream, including the current status of the stream, its Amazon Resource Name
get_records	Retrieves the stream records from a given shard
get_shard_iterator	Returns a shard iterator
list_streams	Returns an array of stream ARNs associated with the current account and endpoint

Examples

```
## Not run:
svc <- dynamodbstreams()
# The following example describes a stream with a given stream ARN.
svc$describe_stream(
  StreamArn = "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/stream/2..."
)

## End(Not run)
```

 elasticache

 Amazon ElastiCache

Description

Amazon ElastiCache is a web service that makes it easier to set up, operate, and scale a distributed cache in the cloud.

With ElastiCache, customers get all of the benefits of a high-performance, in-memory cache with less of the administrative burden involved in launching and managing a distributed cache. The service makes setup, scaling, and cluster failure handling much simpler than in a self-managed cache deployment.

In addition, through integration with Amazon CloudWatch, customers get enhanced visibility into the key performance statistics associated with their cache and can receive alarms if a part of their cache runs hot.

Usage

```
elasticache(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- elasticache(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

[add_tags_to_resource](#)

[authorize_cache_security_group_ingress](#)

[batch_apply_update_action](#)

[batch_stop_update_action](#)

Adds up to 50 cost allocation tags to the named resource

Allows network ingress to a cache security group

Apply the service update

Stop the service update

complete_migration	Complete the migration of data
copy_snapshot	Makes a copy of an existing snapshot
create_cache_cluster	Creates a cluster
create_cache_parameter_group	Creates a new Amazon ElastiCache cache parameter group
create_cache_security_group	Creates a new cache security group
create_cache_subnet_group	Creates a new cache subnet group
create_global_replication_group	Global Datastore for Redis offers fully managed, fast, reliable and secure
create_replication_group	Creates a Redis (cluster mode disabled) or a Redis (cluster mode enabled)
create_snapshot	Creates a copy of an entire cluster or replication group at a specific moment
create_user	For Redis engine version 6
create_user_group	For Redis engine version 6
decrease_node_groups_in_global_replication_group	Decreases the number of node groups in a Global Datastore
decrease_replica_count	Dynamically decreases the number of replicas in a Redis (cluster mode enabled)
delete_cache_cluster	Deletes a previously provisioned cluster
delete_cache_parameter_group	Deletes the specified cache parameter group
delete_cache_security_group	Deletes a cache security group
delete_cache_subnet_group	Deletes a cache subnet group
delete_global_replication_group	Deleting a Global Datastore is a two-step process:
delete_replication_group	Deletes an existing replication group
delete_snapshot	Deletes an existing snapshot
delete_user	For Redis engine version 6
delete_user_group	For Redis engine version 6
describe_cache_clusters	Returns information about all provisioned clusters if no cluster identifier is provided
describe_cache_engine_versions	Returns a list of the available cache engines and their versions
describe_cache_parameter_groups	Returns a list of cache parameter group descriptions
describe_cache_parameters	Returns the detailed parameter list for a particular cache parameter group
describe_cache_security_groups	Returns a list of cache security group descriptions
describe_cache_subnet_groups	Returns a list of cache subnet group descriptions
describe_engine_default_parameters	Returns the default engine and system parameter information for the specified engine
describe_events	Returns events related to clusters, cache security groups, and cache parameter groups
describe_global_replication_groups	Returns information about a particular global replication group
describe_replication_groups	Returns information about a particular replication group
describe_reserved_cache_nodes	Returns information about reserved cache nodes for this account, or about all reserved cache nodes
describe_reserved_cache_nodes_offerings	Lists available reserved cache node offerings
describe_service_updates	Returns details of the service updates
describe_snapshots	Returns information about cluster or replication group snapshots
describe_update_actions	Returns details of the update actions
describe_user_groups	Returns a list of user groups
describe_users	Returns a list of users
disassociate_global_replication_group	Remove a secondary cluster from the Global Datastore using the Global Datastore
failover_global_replication_group	Used to failover the primary region to a selected secondary region
increase_node_groups_in_global_replication_group	Increase the number of node groups in the Global Datastore
increase_replica_count	Dynamically increases the number of replicas in a Redis (cluster mode enabled)
list_allowed_node_type_modifications	Lists all available node types that you can scale your Redis cluster's or replication group's
list_tags_for_resource	Lists all cost allocation tags currently on the named resource
modify_cache_cluster	Modifies the settings for a cluster
modify_cache_parameter_group	Modifies the parameters of a cache parameter group
modify_cache_subnet_group	Modifies an existing cache subnet group

modify_global_replication_group	Modifies the settings for a Global Datastore
modify_replication_group	Modifies the settings for a replication group
modify_replication_group_shard_configuration	Modifies a replication group's shards (node groups) by allowing you to
modify_user	Changes user password(s) and/or access string
modify_user_group	Changes the list of users that belong to the user group
purchase_reserved_cache_nodes_offering	Allows you to purchase a reserved cache node offering
rebalance_slots_in_global_replication_group	Redistribute slots to ensure uniform distribution across existing shards
reboot_cache_cluster	Reboots some, or all, of the cache nodes within a provisioned cluster
remove_tags_from_resource	Removes the tags identified by the TagKeys list from the named resource
reset_cache_parameter_group	Modifies the parameters of a cache parameter group to the engine or sy
revoke_cache_security_group_ingress	Revokes ingress from a cache security group
start_migration	Start the migration of data
test_failover	Represents the input of a TestFailover operation which test automatic f

Examples

```
## Not run:
svc <- elasticache()
svc$add_tags_to_resource(
  Foo = 123
)

## End(Not run)
```

neptune

Amazon Neptune

Description

Amazon Neptune is a fast, reliable, fully-managed graph database service that makes it easy to build and run applications that work with highly connected datasets. The core of Amazon Neptune is a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Amazon Neptune supports popular graph models Property Graph and W3C's RDF, and their respective query languages Apache TinkerPop Gremlin and SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets. Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security.

This interface reference for Amazon Neptune contains documentation for a programming or command line interface you can use to manage Amazon Neptune. Note that Amazon Neptune is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

Usage

```
neptune(config = list())
```

Arguments

config Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- neptune(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_role_to_db_cluster	Associates an Identity and Access Management (IAM) role from an Neptune I
add_source_identifier_to_subscription	Adds a source identifier to an existing event notification subscription
add_tags_to_resource	Adds metadata tags to an Amazon Neptune resource
apply_pending_maintenance_action	Applies a pending maintenance action to a resource (for example, to a DB inst
copy_db_cluster_parameter_group	Copies the specified DB cluster parameter group
copy_db_cluster_snapshot	Copies a snapshot of a DB cluster
copy_db_parameter_group	Copies the specified DB parameter group
create_db_cluster	Creates a new Amazon Neptune DB cluster
create_db_cluster_endpoint	Creates a new custom endpoint and associates it with an Amazon Neptune DB
create_db_cluster_parameter_group	Creates a new DB cluster parameter group
create_db_cluster_snapshot	Creates a snapshot of a DB cluster
create_db_instance	Creates a new DB instance
create_db_parameter_group	Creates a new DB parameter group
create_db_subnet_group	Creates a new DB subnet group
create_event_subscription	Creates an event notification subscription
delete_db_cluster	The DeleteDBCluster action deletes a previously provisioned DB cluster
delete_db_cluster_endpoint	Deletes a custom endpoint and removes it from an Amazon Neptune DB clust
delete_db_cluster_parameter_group	Deletes a specified DB cluster parameter group
delete_db_cluster_snapshot	Deletes a DB cluster snapshot
delete_db_instance	The DeleteDBInstance action deletes a previously provisioned DB instance
delete_db_parameter_group	Deletes a specified DBParameterGroup
delete_db_subnet_group	Deletes a DB subnet group

<code>delete_event_subscription</code>	Deletes an event notification subscription
<code>describe_db_cluster_endpoints</code>	Returns information about endpoints for an Amazon Neptune DB cluster
<code>describe_db_cluster_parameter_groups</code>	Returns a list of DBClusterParameterGroup descriptions
<code>describe_db_cluster_parameters</code>	Returns the detailed parameter list for a particular DB cluster parameter group
<code>describe_db_clusters</code>	Returns information about provisioned DB clusters, and supports pagination
<code>describe_db_cluster_snapshot_attributes</code>	Returns a list of DB cluster snapshot attribute names and values for a manual I
<code>describe_db_cluster_snapshots</code>	Returns information about DB cluster snapshots
<code>describe_db_engine_versions</code>	Returns a list of the available DB engines
<code>describe_db_instances</code>	Returns information about provisioned instances, and supports pagination
<code>describe_db_parameter_groups</code>	Returns a list of DBParameterGroup descriptions
<code>describe_db_parameters</code>	Returns the detailed parameter list for a particular DB parameter group
<code>describe_db_subnet_groups</code>	Returns a list of DBSubnetGroup descriptions
<code>describe_engine_default_cluster_parameters</code>	Returns the default engine and system parameter information for the cluster da
<code>describe_engine_default_parameters</code>	Returns the default engine and system parameter information for the specified
<code>describe_event_categories</code>	Displays a list of categories for all event source types, or, if specified, for a spe
<code>describe_events</code>	Returns events related to DB instances, DB security groups, DB snapshots, and
<code>describe_event_subscriptions</code>	Lists all the subscription descriptions for a customer account
<code>describe_orderable_db_instance_options</code>	Returns a list of orderable DB instance options for the specified engine
<code>describe_pending_maintenance_actions</code>	Returns a list of resources (for example, DB instances) that have at least one p
<code>describe_valid_db_instance_modifications</code>	You can call DescribeValidDBInstanceModifications to learn what modificatio
<code>failover_db_cluster</code>	Forces a failover for a DB cluster
<code>list_tags_for_resource</code>	Lists all tags on an Amazon Neptune resource
<code>modify_db_cluster</code>	Modify a setting for a DB cluster
<code>modify_db_cluster_endpoint</code>	Modifies the properties of an endpoint in an Amazon Neptune DB cluster
<code>modify_db_cluster_parameter_group</code>	Modifies the parameters of a DB cluster parameter group
<code>modify_db_cluster_snapshot_attribute</code>	Adds an attribute and values to, or removes an attribute and values from, a mar
<code>modify_db_instance</code>	Modifies settings for a DB instance
<code>modify_db_parameter_group</code>	Modifies the parameters of a DB parameter group
<code>modify_db_subnet_group</code>	Modifies an existing DB subnet group
<code>modify_event_subscription</code>	Modifies an existing event notification subscription
<code>promote_read_replica_db_cluster</code>	Not supported
<code>reboot_db_instance</code>	You might need to reboot your DB instance, usually for maintenance reasons
<code>remove_role_from_db_cluster</code>	Disassociates an Identity and Access Management (IAM) role from a DB clus
<code>remove_source_identifier_from_subscription</code>	Removes a source identifier from an existing event notification subscription
<code>remove_tags_from_resource</code>	Removes metadata tags from an Amazon Neptune resource
<code>reset_db_cluster_parameter_group</code>	Modifies the parameters of a DB cluster parameter group to the default value
<code>reset_db_parameter_group</code>	Modifies the parameters of a DB parameter group to the engine/system default
<code>restore_db_cluster_from_snapshot</code>	Creates a new DB cluster from a DB snapshot or DB cluster snapshot
<code>restore_db_cluster_to_point_in_time</code>	Restores a DB cluster to an arbitrary point in time
<code>start_db_cluster</code>	Starts an Amazon Neptune DB cluster that was stopped using the AWS consol
<code>stop_db_cluster</code>	Stops an Amazon Neptune DB cluster

Examples

```
## Not run:
svc <- neptune()
svc$add_role_to_db_cluster(
```

```
    Foo = 123
)

## End(Not run)
```

rds

Amazon Relational Database Service

Description

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizeable capacity for an industry-standard relational database and manages common database administration tasks, freeing up developers to focus on what makes their applications and businesses unique.

Amazon RDS gives you access to the capabilities of a MySQL, MariaDB, PostgreSQL, Microsoft SQL Server, Oracle, or Amazon Aurora database server. These capabilities mean that the code, applications, and tools you already use today with your existing databases work with Amazon RDS without modification. Amazon RDS automatically backs up your database and maintains the database software that powers your DB instance. Amazon RDS is flexible: you can scale your DB instance's compute resources and storage capacity to meet your application's demand. As with all Amazon Web Services, there are no up-front investments, and you pay only for the resources you use.

This interface reference for Amazon RDS contains documentation for a programming or command line interface you can use to manage Amazon RDS. Amazon RDS is asynchronous, which means that some interfaces might require techniques such as polling or callback functions to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a command is applied immediately, on the next instance reboot, or during the maintenance window. The reference structure is as follows, and we list following some related topics from the user guide.

Amazon RDS API Reference

- For the alphabetical list of API actions, see [API Actions](#).
- For the alphabetical list of data types, see [Data Types](#).
- For a list of common query parameters, see [Common Parameters](#).
- For descriptions of the error codes, see [Common Errors](#).

Amazon RDS User Guide

- For a summary of the Amazon RDS interfaces, see [Available RDS Interfaces](#).
- For more information about how to use the Query API, see [Using the Query API](#).

Usage

```
rds(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- rds(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_role_to_db_cluster	Associates an Identity and Access Management (IAM) role from an Amazon IAM role to an Amazon RDS DB cluster.
add_role_to_db_instance	Associates an AWS Identity and Access Management (IAM) role with an Amazon RDS DB instance.
add_source_identifier_to_subscription	Adds a source identifier to an existing RDS event notification subscription.
add_tags_to_resource	Adds metadata tags to an Amazon RDS resource.
apply_pending_maintenance_action	Applies a pending maintenance action to a resource (for example, to a DB instance).
authorize_db_security_group_ingress	Enables ingress to a DBSecurityGroup using one of two forms of authorization.
backtrack_db_cluster	Backtracks a DB cluster to a specific time, without creating a new DB cluster.
build_auth_token	Return an authentication token for a database connection.
cancel_export_task	Cancels an export task in progress that is exporting a snapshot to Amazon S3.
copy_db_cluster_parameter_group	Copies the specified DB cluster parameter group.
copy_db_cluster_snapshot	Copies a snapshot of a DB cluster.
copy_db_parameter_group	Copies the specified DB parameter group.
copy_db_snapshot	Copies the specified DB snapshot.
copy_option_group	Copies the specified option group.
create_custom_availability_zone	Creates a custom Availability Zone (AZ).
create_db_cluster	Creates a new Amazon Aurora DB cluster.
create_db_cluster_endpoint	Creates a new custom endpoint and associates it with an Amazon Aurora DB cluster.
create_db_cluster_parameter_group	Creates a new DB cluster parameter group.
create_db_cluster_snapshot	Creates a snapshot of a DB cluster.
create_db_instance	Creates a new DB instance.
create_db_instance_read_replica	Creates a new DB instance that acts as a read replica for an existing source instance.
create_db_parameter_group	Creates a new DB parameter group.
create_db_proxy	Creates a new DB proxy.
create_db_security_group	Creates a new DB security group.
create_db_snapshot	Creates a snapshot of a DB instance.
create_db_subnet_group	Creates a new DB subnet group.

create_event_subscription	Creates an RDS event notification subscription
create_global_cluster	Creates an Aurora global database spread across multiple AWS Regions
create_option_group	Creates a new option group
delete_custom_availability_zone	Deletes a custom Availability Zone (AZ)
delete_db_cluster	The DeleteDBCluster action deletes a previously provisioned DB cluster
delete_db_cluster_endpoint	Deletes a custom endpoint and removes it from an Amazon Aurora DB cluster
delete_db_cluster_parameter_group	Deletes a specified DB cluster parameter group
delete_db_cluster_snapshot	Deletes a DB cluster snapshot
delete_db_instance	The DeleteDBInstance action deletes a previously provisioned DB instance
delete_db_instance_automated_backup	Deletes automated backups using the DbiResourceId value of the source instance
delete_db_parameter_group	Deletes a specified DB parameter group
delete_db_proxy	Deletes an existing proxy
delete_db_security_group	Deletes a DB security group
delete_db_snapshot	Deletes a DB snapshot
delete_db_subnet_group	Deletes a DB subnet group
delete_event_subscription	Deletes an RDS event notification subscription
delete_global_cluster	Deletes a global database cluster
delete_installation_media	Deletes the installation medium for a DB engine that requires an on-premise installation
delete_option_group	Deletes an existing option group
deregister_db_proxy_targets	Remove the association between one or more DBProxyTarget data structures and a DB proxy
describe_account_attributes	Lists all of the attributes for a customer account
describe_certificates	Lists the set of CA certificates provided by Amazon RDS for this AWS account
describe_custom_availability_zones	Returns information about custom Availability Zones (AZs)
describe_db_cluster_backtracks	Returns information about backtracks for a DB cluster
describe_db_cluster_endpoints	Returns information about endpoints for an Amazon Aurora DB cluster
describe_db_cluster_parameter_groups	Returns a list of DBClusterParameterGroup descriptions
describe_db_cluster_parameters	Returns the detailed parameter list for a particular DB cluster parameter group
describe_db_clusters	Returns information about provisioned Aurora DB clusters
describe_db_cluster_snapshot_attributes	Returns a list of DB cluster snapshot attribute names and values for a manual DB cluster snapshot
describe_db_cluster_snapshots	Returns information about DB cluster snapshots
describe_db_engine_versions	Returns a list of the available DB engines
describe_db_instance_automated_backups	Displays backups for both current and deleted instances
describe_db_instances	Returns information about provisioned RDS instances
describe_db_log_files	Returns a list of DB log files for the DB instance
describe_db_parameter_groups	Returns a list of DBParameterGroup descriptions
describe_db_parameters	Returns the detailed parameter list for a particular DB parameter group
describe_db_proxies	Returns information about DB proxies
describe_db_proxy_target_groups	Returns information about DB proxy target groups, represented by DBProxyTargetGroup objects
describe_db_proxy_targets	Returns information about DBProxyTarget objects
describe_db_security_groups	Returns a list of DBSecurityGroup descriptions
describe_db_snapshot_attributes	Returns a list of DB snapshot attribute names and values for a manual DB snapshot
describe_db_snapshots	Returns information about DB snapshots
describe_db_subnet_groups	Returns a list of DBSubnetGroup descriptions
describe_engine_default_cluster_parameters	Returns the default engine and system parameter information for the cluster
describe_engine_default_parameters	Returns the default engine and system parameter information for the specified engine
describe_event_categories	Displays a list of categories for all event source types, or, if specified, for a particular event source type
describe_events	Returns events related to DB instances, DB clusters, DB parameter groups, or DB snapshots
describe_event_subscriptions	Lists all the subscription descriptions for a customer account

<code>describe_export_tasks</code>	Returns information about a snapshot export to Amazon S3
<code>describe_global_clusters</code>	Returns information about Aurora global database clusters
<code>describe_installation_media</code>	Describes the available installation media for a DB engine that requires a
<code>describe_option_group_options</code>	Describes all available options
<code>describe_option_groups</code>	Describes the available option groups
<code>describe_orderable_db_instance_options</code>	Returns a list of orderable DB instance options for the specified engine
<code>describe_pending_maintenance_actions</code>	Returns a list of resources (for example, DB instances) that have at least
<code>describe_reserved_db_instances</code>	Returns information about reserved DB instances for this account, or abo
<code>describe_reserved_db_instances_offerings</code>	Lists available reserved DB instance offerings
<code>describe_source_regions</code>	Returns a list of the source AWS Regions where the current AWS Region
<code>describe_valid_db_instance_modifications</code>	You can call DescribeValidDBInstanceModifications to learn what modifi
<code>download_db_log_file_portion</code>	Downloads all or a portion of the specified log file, up to 1 MB in size
<code>failover_db_cluster</code>	Forces a failover for a DB cluster
<code>import_installation_media</code>	Imports the installation media for a DB engine that requires an on-premis
<code>list_tags_for_resource</code>	Lists all tags on an Amazon RDS resource
<code>modify_certificates</code>	Override the system-default Secure Sockets Layer/Transport Layer Secur
<code>modify_current_db_cluster_capacity</code>	Set the capacity of an Aurora Serverless DB cluster to a specific value
<code>modify_db_cluster</code>	Modify a setting for an Amazon Aurora DB cluster
<code>modify_db_cluster_endpoint</code>	Modifies the properties of an endpoint in an Amazon Aurora DB cluster
<code>modify_db_cluster_parameter_group</code>	Modifies the parameters of a DB cluster parameter group
<code>modify_db_cluster_snapshot_attribute</code>	Adds an attribute and values to, or removes an attribute and values from,
<code>modify_db_instance</code>	Modifies settings for a DB instance
<code>modify_db_parameter_group</code>	Modifies the parameters of a DB parameter group
<code>modify_db_proxy</code>	Changes the settings for an existing DB proxy
<code>modify_db_proxy_target_group</code>	Modifies the properties of a DBProxyTargetGroup
<code>modify_db_snapshot</code>	Updates a manual DB snapshot with a new engine version
<code>modify_db_snapshot_attribute</code>	Adds an attribute and values to, or removes an attribute and values from,
<code>modify_db_subnet_group</code>	Modifies an existing DB subnet group
<code>modify_event_subscription</code>	Modifies an existing RDS event notification subscription
<code>modify_global_cluster</code>	Modify a setting for an Amazon Aurora global cluster
<code>modify_option_group</code>	Modifies an existing option group
<code>promote_read_replica</code>	Promotes a read replica DB instance to a standalone DB instance
<code>promote_read_replica_db_cluster</code>	Promotes a read replica DB cluster to a standalone DB cluster
<code>purchase_reserved_db_instances_offering</code>	Purchases a reserved DB instance offering
<code>reboot_db_instance</code>	You might need to reboot your DB instance, usually for maintenance rea
<code>register_db_proxy_targets</code>	Associate one or more DBProxyTarget data structures with a DBProxyTa
<code>remove_from_global_cluster</code>	Detaches an Aurora secondary cluster from an Aurora global database cl
<code>remove_role_from_db_cluster</code>	Disassociates an AWS Identity and Access Management (IAM) role from
<code>remove_role_from_db_instance</code>	Disassociates an AWS Identity and Access Management (IAM) role from
<code>remove_source_identifier_from_subscription</code>	Removes a source identifier from an existing RDS event notification sub
<code>remove_tags_from_resource</code>	Removes metadata tags from an Amazon RDS resource
<code>reset_db_cluster_parameter_group</code>	Modifies the parameters of a DB cluster parameter group to the default v
<code>reset_db_parameter_group</code>	Modifies the parameters of a DB parameter group to the engine/system d
<code>restore_db_cluster_from_s3</code>	Creates an Amazon Aurora DB cluster from MySQL data stored in an A
<code>restore_db_cluster_from_snapshot</code>	Creates a new DB cluster from a DB snapshot or DB cluster snapshot
<code>restore_db_cluster_to_point_in_time</code>	Restores a DB cluster to an arbitrary point in time
<code>restore_db_instance_from_db_snapshot</code>	Creates a new DB instance from a DB snapshot
<code>restore_db_instance_from_s3</code>	Amazon Relational Database Service (Amazon RDS) supports importing

restore_db_instance_to_point_in_time	Restores a DB instance to an arbitrary point in time
revoke_db_security_group_ingress	Revokes ingress from a DBSecurityGroup for previously authorized IP ranges
start_activity_stream	Starts a database activity stream to monitor activity on the database
start_db_cluster	Starts an Amazon Aurora DB cluster that was stopped using the AWS console
start_db_instance	Starts an Amazon RDS DB instance that was stopped using the AWS console
start_db_instance_automated_backups_replication	Enables replication of automated backups to a different AWS Region
start_export_task	Starts an export of a snapshot to Amazon S3
stop_activity_stream	Stops a database activity stream that was started using the AWS console
stop_db_cluster	Stops an Amazon Aurora DB cluster
stop_db_instance	Stops an Amazon RDS DB instance
stop_db_instance_automated_backups_replication	Stops automated backup replication for a DB instance

Examples

```
## Not run:
svc <- rds()
svc$add_role_to_db_cluster(
  Foo = 123
)

## End(Not run)
```

rdsdataservice	<i>AWS RDS DataService</i>
----------------	----------------------------

Description

Amazon RDS Data Service

Amazon RDS provides an HTTP endpoint to run SQL statements on an Amazon Aurora Serverless DB cluster. To run these statements, you work with the Data Service API.

For more information about the Data Service API, see [Using the Data API for Aurora Serverless](#) in the *Amazon Aurora User Guide*.

If you have questions or comments related to the Data API, send email to Rds-data-api-feedback@amazon.com.

Usage

```
rdsdataservice(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```

svc <- rdsdataservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)

```

Operations

batch_execute_statement	Runs a batch SQL statement over an array of data
begin_transaction	Starts a SQL transaction
commit_transaction	Ends a SQL transaction started with the BeginTransaction operation and commits the changes
execute_sql	Runs one or more SQL statements
execute_statement	Runs a SQL statement against a database
rollback_transaction	Performs a rollback of a transaction

Examples

```

## Not run:
svc <- rdsdataservice()
svc$batch_execute_statement(
  Foo = 123
)

## End(Not run)

```

redshift

Amazon Redshift

Description**Overview**

This is an interface reference for Amazon Redshift. It contains documentation for one of the programming or command line interfaces you can use to manage Amazon Redshift clusters. Note that

Amazon Redshift is asynchronous, which means that some interfaces may require techniques, such as polling or asynchronous callback handlers, to determine when a command has been applied. In this reference, the parameter descriptions indicate whether a change is applied immediately, on the next instance reboot, or during the next maintenance window. For a summary of the Amazon Redshift cluster management interfaces, go to [Using the Amazon Redshift Management Interfaces](#).

Amazon Redshift manages all the work of setting up, operating, and scaling a data warehouse: provisioning capacity, monitoring and backing up the cluster, and applying patches and upgrades to the Amazon Redshift engine. You can focus on using your data to acquire new insights for your business and customers.

If you are a first-time user of Amazon Redshift, we recommend that you begin by reading the [Amazon Redshift Getting Started Guide](#).

If you are a database developer, the [Amazon Redshift Database Developer Guide](#) explains how to design, build, query, and maintain the databases that make up your data warehouse.

Usage

```
redshift(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- redshift(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

accept_reserved_node_exchange	Exchanges a DC1 Reserved Node for a DC2 Reserved Node with no changes to the
authorize_cluster_security_group_ingress	Adds an inbound (ingress) rule to an Amazon Redshift security group
authorize_snapshot_access	Authorizes the specified AWS customer account to restore the specified snapshot
batch_delete_cluster_snapshots	Deletes a set of cluster snapshots
batch_modify_cluster_snapshots	Modifies the settings for a set of cluster snapshots
cancel_resize	Cancels a resize operation for a cluster
copy_cluster_snapshot	Copies the specified automated cluster snapshot to a new manual cluster snapshot

<code>create_cluster</code>	Creates a new cluster with the specified parameters
<code>create_cluster_parameter_group</code>	Creates an Amazon Redshift parameter group
<code>create_cluster_security_group</code>	Creates a new Amazon Redshift security group
<code>create_cluster_snapshot</code>	Creates a manual snapshot of the specified cluster
<code>create_cluster_subnet_group</code>	Creates a new Amazon Redshift subnet group
<code>create_event_subscription</code>	Creates an Amazon Redshift event notification subscription
<code>create_hsm_client_certificate</code>	Creates an HSM client certificate that an Amazon Redshift cluster will use to connect to an Amazon Key Management Service (KMS) key
<code>create_hsm_configuration</code>	Creates an HSM configuration that contains the information required by an Amazon Redshift cluster to connect to an Amazon Key Management Service (KMS) key
<code>create_scheduled_action</code>	Creates a scheduled action
<code>create_snapshot_copy_grant</code>	Creates a snapshot copy grant that permits Amazon Redshift to use a customer managed KMS key to encrypt snapshots
<code>create_snapshot_schedule</code>	Create a snapshot schedule that can be associated to a cluster and which overrides the default snapshot schedule
<code>create_tags</code>	Adds tags to a cluster
<code>create_usage_limit</code>	Creates a usage limit for a specified Amazon Redshift feature on a cluster
<code>delete_cluster</code>	Deletes a previously provisioned cluster without its final snapshot being created
<code>delete_cluster_parameter_group</code>	Deletes a specified Amazon Redshift parameter group
<code>delete_cluster_security_group</code>	Deletes an Amazon Redshift security group
<code>delete_cluster_snapshot</code>	Deletes the specified manual snapshot
<code>delete_cluster_subnet_group</code>	Deletes the specified cluster subnet group
<code>delete_event_subscription</code>	Deletes an Amazon Redshift event notification subscription
<code>delete_hsm_client_certificate</code>	Deletes the specified HSM client certificate
<code>delete_hsm_configuration</code>	Deletes the specified Amazon Redshift HSM configuration
<code>delete_scheduled_action</code>	Deletes a scheduled action
<code>delete_snapshot_copy_grant</code>	Deletes the specified snapshot copy grant
<code>delete_snapshot_schedule</code>	Deletes a snapshot schedule
<code>delete_tags</code>	Deletes tags from a resource
<code>delete_usage_limit</code>	Deletes a usage limit from a cluster
<code>describe_account_attributes</code>	Returns a list of attributes attached to an account
<code>describe_cluster_db_revisions</code>	Returns an array of ClusterDbRevision objects
<code>describe_cluster_parameter_groups</code>	Returns a list of Amazon Redshift parameter groups, including parameter groups that are associated with clusters
<code>describe_cluster_parameters</code>	Returns a detailed list of parameters contained within the specified Amazon Redshift parameter group
<code>describe_clusters</code>	Returns properties of provisioned clusters including general cluster properties, cluster status, and cluster configuration
<code>describe_cluster_security_groups</code>	Returns information about Amazon Redshift security groups
<code>describe_cluster_snapshots</code>	Returns one or more snapshot objects, which contain metadata about your cluster snapshots
<code>describe_cluster_subnet_groups</code>	Returns one or more cluster subnet group objects, which contain metadata about your cluster subnet groups
<code>describe_cluster_tracks</code>	Returns a list of all the available maintenance tracks
<code>describe_cluster_versions</code>	Returns descriptions of the available Amazon Redshift cluster versions
<code>describe_default_cluster_parameters</code>	Returns a list of parameter settings for the specified parameter group family
<code>describe_event_categories</code>	Displays a list of event categories for all event source types, or for a specified source type
<code>describe_events</code>	Returns events related to clusters, security groups, snapshots, and parameter groups
<code>describe_event_subscriptions</code>	Lists descriptions of all the Amazon Redshift event notification subscriptions for a specified source type
<code>describe_hsm_client_certificates</code>	Returns information about the specified HSM client certificate
<code>describe_hsm_configurations</code>	Returns information about the specified Amazon Redshift HSM configuration
<code>describe_logging_status</code>	Describes whether information, such as queries and connection attempts, is being logged to an Amazon S3 bucket
<code>describe_node_configuration_options</code>	Returns properties of possible node configurations such as node type, number of nodes, and node configuration options
<code>describe_orderable_cluster_options</code>	Returns a list of orderable cluster options
<code>describe_reserved_node_offerings</code>	Returns a list of the available reserved node offerings by Amazon Redshift with their respective attributes
<code>describe_reserved_nodes</code>	Returns the descriptions of the reserved nodes
<code>describe_resize</code>	Returns information about the last resize operation for the specified cluster

<code>describe_scheduled_actions</code>	Describes properties of scheduled actions
<code>describe_snapshot_copy_grants</code>	Returns a list of snapshot copy grants owned by the AWS account in the destination
<code>describe_snapshot_schedules</code>	Returns a list of snapshot schedules
<code>describe_storage</code>	Returns account level backups storage size and provisional storage
<code>describe_table_restore_status</code>	Lists the status of one or more table restore requests made using the RestoreTable
<code>describe_tags</code>	Returns a list of tags
<code>describe_usage_limits</code>	Shows usage limits on a cluster
<code>disable_logging</code>	Stops logging information, such as queries and connection attempts, for the speci
<code>disable_snapshot_copy</code>	Disables the automatic copying of snapshots from one region to another region fo
<code>enable_logging</code>	Starts logging information, such as queries and connection attempts, for the speci
<code>enable_snapshot_copy</code>	Enables the automatic copy of snapshots from one region to another region for a s
<code>get_cluster_credentials</code>	Returns a database user name and temporary password with temporary authorizat
<code>get_reserved_node_exchange_offerings</code>	Returns an array of DC2 ReservedNodeOfferings that matches the payment type,
<code>modify_cluster</code>	Modifies the settings for a cluster
<code>modify_cluster_db_revision</code>	Modifies the database revision of a cluster
<code>modify_cluster_iam_roles</code>	Modifies the list of AWS Identity and Access Management (IAM) roles that can b
<code>modify_cluster_maintenance</code>	Modifies the maintenance settings of a cluster
<code>modify_cluster_parameter_group</code>	Modifies the parameters of a parameter group
<code>modify_cluster_snapshot</code>	Modifies the settings for a snapshot
<code>modify_cluster_snapshot_schedule</code>	Modifies a snapshot schedule for a cluster
<code>modify_cluster_subnet_group</code>	Modifies a cluster subnet group to include the specified list of VPC subnets
<code>modify_event_subscription</code>	Modifies an existing Amazon Redshift event notification subscription
<code>modify_scheduled_action</code>	Modifies a scheduled action
<code>modify_snapshot_copy_retention_period</code>	Modifies the number of days to retain snapshots in the destination AWS Region a
<code>modify_snapshot_schedule</code>	Modifies a snapshot schedule
<code>modify_usage_limit</code>	Modifies a usage limit in a cluster
<code>pause_cluster</code>	Pauses a cluster
<code>purchase_reserved_node_offering</code>	Allows you to purchase reserved nodes
<code>reboot_cluster</code>	Reboots a cluster
<code>reset_cluster_parameter_group</code>	Sets one or more parameters of the specified parameter group to their default valu
<code>resize_cluster</code>	Changes the size of the cluster
<code>restore_from_cluster_snapshot</code>	Creates a new cluster from a snapshot
<code>restore_table_from_cluster_snapshot</code>	Creates a new table from a table in an Amazon Redshift cluster snapshot
<code>resume_cluster</code>	Resumes a paused cluster
<code>revoke_cluster_security_group_ingress</code>	Revokes an ingress rule in an Amazon Redshift security group for a previously au
<code>revoke_snapshot_access</code>	Removes the ability of the specified AWS customer account to restore the specifie
<code>rotate_encryption_key</code>	Rotates the encryption keys for a cluster

Examples

```
## Not run:
svc <- redshift()
svc$accept_reserved_node_exchange(
  Foo = 123
)

## End(Not run)
```

`simpledb`*Amazon SimpleDB*

Description

Amazon SimpleDB is a web service providing the core database functions of data indexing and querying in the cloud. By offloading the time and effort associated with building and operating a web-scale database, SimpleDB provides developers the freedom to focus on application development.

A traditional, clustered relational database requires a sizable upfront capital outlay, is complex to design, and often requires extensive and repetitive database administration. Amazon SimpleDB is dramatically simpler, requiring no schema, automatically indexing your data and providing a simple API for storage and access. This approach eliminates the administrative burden of data modeling, index maintenance, and performance tuning. Developers gain access to this functionality within Amazon's proven computing environment, are able to scale instantly, and pay only for what they use.

Visit <http://aws.amazon.com/simpledb/> for more information.

Usage

```
simpledb(config = list())
```

Arguments

`config` Optional configuration of credentials, endpoint, and/or region.

Service syntax

```
svc <- simpledb(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string"  
    ),  
    endpoint = "string",  
    region = "string"  
  )  
)
```

Operations

batch_delete_attributes	Performs multiple DeleteAttributes operations in a single call, which reduces round trips and latency
batch_put_attributes	The BatchPutAttributes operation creates or replaces attributes within one or more items
create_domain	The CreateDomain operation creates a new domain
delete_attributes	Deletes one or more attributes associated with an item
delete_domain	The DeleteDomain operation deletes a domain
domain_metadata	Returns information about the domain, including when the domain was created, the number of items
get_attributes	Returns all of the attributes associated with the specified item
list_domains	The ListDomains operation lists all domains associated with the Access Key ID
put_attributes	The PutAttributes operation creates or replaces attributes in an item
select	The Select operation returns a set of attributes for ItemNames that match the select expression

Examples

```
## Not run:
svc <- simpledb()
svc$batch_delete_attributes(
  Foo = 123
)

## End(Not run)
```

Index

accept_reserved_node_exchange, [21](#)
add_role_to_db_cluster, [13](#), [16](#)
add_role_to_db_instance, [16](#)
add_source_identifier_to_subscription,
[13](#), [16](#)
add_tags_to_resource, [4](#), [10](#), [13](#), [16](#)
apply_pending_maintenance_action, [4](#), [13](#),
[16](#)
authorize_cache_security_group_ingress,
[10](#)
authorize_cluster_security_group_ingress,
[21](#)
authorize_db_security_group_ingress,
[16](#)
authorize_snapshot_access, [21](#)

backtrack_db_cluster, [16](#)
batch_apply_update_action, [10](#)
batch_delete_attributes, [25](#)
batch_delete_cluster_snapshots, [21](#)
batch_execute_statement, [6](#), [20](#)
batch_get_item, [6](#)
batch_modify_cluster_snapshots, [21](#)
batch_put_attributes, [25](#)
batch_stop_update_action, [10](#)
batch_write_item, [6](#)
begin_transaction, [20](#)
build_auth_token, [16](#)

cancel_export_task, [16](#)
cancel_resize, [21](#)
commit_transaction, [20](#)
complete_migration, [11](#)
copy_cluster_snapshot, [21](#)
copy_db_cluster_parameter_group, [4](#), [13](#),
[16](#)
copy_db_cluster_snapshot, [4](#), [13](#), [16](#)
copy_db_parameter_group, [13](#), [16](#)
copy_db_snapshot, [16](#)
copy_option_group, [16](#)

copy_snapshot, [11](#)
create_backup, [6](#)
create_cache_cluster, [11](#)
create_cache_parameter_group, [11](#)
create_cache_security_group, [11](#)
create_cache_subnet_group, [11](#)
create_cluster, [3](#), [22](#)
create_cluster_parameter_group, [22](#)
create_cluster_security_group, [22](#)
create_cluster_snapshot, [22](#)
create_cluster_subnet_group, [22](#)
create_custom_availability_zone, [16](#)
create_db_cluster, [4](#), [13](#), [16](#)
create_db_cluster_endpoint, [13](#), [16](#)
create_db_cluster_parameter_group, [4](#),
[13](#), [16](#)
create_db_cluster_snapshot, [4](#), [13](#), [16](#)
create_db_instance, [4](#), [13](#), [16](#)
create_db_instance_read_replica, [16](#)
create_db_parameter_group, [13](#), [16](#)
create_db_proxy, [16](#)
create_db_security_group, [16](#)
create_db_snapshot, [16](#)
create_db_subnet_group, [4](#), [13](#), [16](#)
create_domain, [25](#)
create_event_subscription, [13](#), [17](#), [22](#)
create_global_cluster, [17](#)
create_global_replication_group, [11](#)
create_global_table, [6](#)
create_hsm_client_certificate, [22](#)
create_hsm_configuration, [22](#)
create_option_group, [17](#)
create_parameter_group, [3](#)
create_replication_group, [11](#)
create_scheduled_action, [22](#)
create_snapshot, [11](#)
create_snapshot_copy_grant, [22](#)
create_snapshot_schedule, [22](#)
create_subnet_group, [3](#)

- create_table, [6](#)
- create_tags, [22](#)
- create_usage_limit, [22](#)
- create_user, [11](#)
- create_user_group, [11](#)

- dax, [2](#)
- decrease_node_groups_in_global_replication_group, [11](#)
- decrease_replica_count, [11](#)
- decrease_replication_factor, [3](#)
- delete_attributes, [25](#)
- delete_backup, [7](#)
- delete_cache_cluster, [11](#)
- delete_cache_parameter_group, [11](#)
- delete_cache_security_group, [11](#)
- delete_cache_subnet_group, [11](#)
- delete_cluster, [3](#), [22](#)
- delete_cluster_parameter_group, [22](#)
- delete_cluster_security_group, [22](#)
- delete_cluster_snapshot, [22](#)
- delete_cluster_subnet_group, [22](#)
- delete_custom_availability_zone, [17](#)
- delete_db_cluster, [4](#), [13](#), [17](#)
- delete_db_cluster_endpoint, [13](#), [17](#)
- delete_db_cluster_parameter_group, [4](#), [13](#), [17](#)
- delete_db_cluster_snapshot, [4](#), [13](#), [17](#)
- delete_db_instance, [4](#), [13](#), [17](#)
- delete_db_instance_automated_backup, [17](#)
- delete_db_parameter_group, [13](#), [17](#)
- delete_db_proxy, [17](#)
- delete_db_security_group, [17](#)
- delete_db_snapshot, [17](#)
- delete_db_subnet_group, [5](#), [13](#), [17](#)
- delete_domain, [25](#)
- delete_event_subscription, [14](#), [17](#), [22](#)
- delete_global_cluster, [17](#)
- delete_global_replication_group, [11](#)
- delete_hsm_client_certificate, [22](#)
- delete_hsm_configuration, [22](#)
- delete_installation_media, [17](#)
- delete_item, [7](#)
- delete_option_group, [17](#)
- delete_parameter_group, [3](#)
- delete_replication_group, [11](#)
- delete_scheduled_action, [22](#)
- delete_snapshot, [11](#)
- delete_snapshot_copy_grant, [22](#)
- delete_snapshot_schedule, [22](#)
- delete_subnet_group, [3](#)
- delete_table, [7](#)
- delete_tags, [22](#)
- delete_usage_limit, [22](#)
- delete_user, [11](#)
- delete_user_group, [11](#)
- deregister_db_proxy_targets, [17](#)
- describe_account_attributes, [17](#), [22](#)
- describe_backup, [7](#)
- describe_cache_clusters, [11](#)
- describe_cache_engine_versions, [11](#)
- describe_cache_parameter_groups, [11](#)
- describe_cache_parameters, [11](#)
- describe_cache_security_groups, [11](#)
- describe_cache_subnet_groups, [11](#)
- describe_certificates, [5](#), [17](#)
- describe_cluster_db_revisions, [22](#)
- describe_cluster_parameter_groups, [22](#)
- describe_cluster_parameters, [22](#)
- describe_cluster_security_groups, [22](#)
- describe_cluster_snapshots, [22](#)
- describe_cluster_subnet_groups, [22](#)
- describe_cluster_tracks, [22](#)
- describe_cluster_versions, [22](#)
- describe_clusters, [3](#), [22](#)
- describe_continuous_backups, [7](#)
- describe_contributor_insights, [7](#)
- describe_custom_availability_zones, [17](#)
- describe_db_cluster_backtracks, [17](#)
- describe_db_cluster_endpoints, [14](#), [17](#)
- describe_db_cluster_parameter_groups, [5](#), [14](#), [17](#)
- describe_db_cluster_parameters, [5](#), [14](#), [17](#)
- describe_db_cluster_snapshot_attributes, [5](#), [14](#), [17](#)
- describe_db_cluster_snapshots, [5](#), [14](#), [17](#)
- describe_db_clusters, [5](#), [14](#), [17](#)
- describe_db_engine_versions, [5](#), [14](#), [17](#)
- describe_db_instance_automated_backups, [17](#)
- describe_db_instances, [5](#), [14](#), [17](#)
- describe_db_log_files, [17](#)
- describe_db_parameter_groups, [14](#), [17](#)
- describe_db_parameters, [14](#), [17](#)
- describe_db_proxies, [17](#)

- describe_db_proxy_target_groups, [17](#)
- describe_db_proxy_targets, [17](#)
- describe_db_security_groups, [17](#)
- describe_db_snapshot_attributes, [17](#)
- describe_db_snapshots, [17](#)
- describe_db_subnet_groups, [5](#), [14](#), [17](#)
- describe_default_cluster_parameters, [22](#)
- describe_default_parameters, [3](#)
- describe_endpoints, [7](#)
- describe_engine_default_cluster_parameters, [5](#), [14](#), [17](#)
- describe_engine_default_parameters, [11](#), [14](#), [17](#)
- describe_event_categories, [5](#), [14](#), [17](#), [22](#)
- describe_event_subscriptions, [14](#), [17](#), [22](#)
- describe_events, [3](#), [5](#), [11](#), [14](#), [17](#), [22](#)
- describe_export, [7](#)
- describe_export_tasks, [18](#)
- describe_global_clusters, [18](#)
- describe_global_replication_groups, [11](#)
- describe_global_table, [7](#)
- describe_global_table_settings, [7](#)
- describe_hsm_client_certificates, [22](#)
- describe_hsm_configurations, [22](#)
- describe_installation_media, [18](#)
- describe_kinesis_streaming_destination, [7](#)
- describe_limits, [7](#)
- describe_logging_status, [22](#)
- describe_node_configuration_options, [22](#)
- describe_option_group_options, [18](#)
- describe_option_groups, [18](#)
- describe_orderable_cluster_options, [22](#)
- describe_orderable_db_instance_options, [5](#), [14](#), [18](#)
- describe_parameter_groups, [3](#)
- describe_parameters, [3](#)
- describe_pending_maintenance_actions, [5](#), [14](#), [18](#)
- describe_replication_groups, [11](#)
- describe_reserved_cache_nodes, [11](#)
- describe_reserved_cache_nodes_offerings, [11](#)
- describe_reserved_db_instances, [18](#)
- describe_reserved_db_instances_offerings, [18](#)
- describe_reserved_node_offerings, [22](#)
- describe_reserved_nodes, [22](#)
- describe_resize, [22](#)
- describe_scheduled_actions, [23](#)
- describe_service_updates, [11](#)
- describe_snapshot_copy_grants, [23](#)
- describe_snapshot_schedules, [23](#)
- describe_snapshots, [11](#)
- describe_source_regions, [18](#)
- describe_storage, [23](#)
- describe_stream, [9](#)
- describe_subnet_groups, [3](#)
- describe_table, [7](#)
- describe_table_replica_auto_scaling, [7](#)
- describe_table_restore_status, [23](#)
- describe_tags, [23](#)
- describe_time_to_live, [7](#)
- describe_update_actions, [11](#)
- describe_usage_limits, [23](#)
- describe_user_groups, [11](#)
- describe_users, [11](#)
- describe_valid_db_instance_modifications, [14](#), [18](#)
- disable_kinesis_streaming_destination, [7](#)
- disable_logging, [23](#)
- disable_snapshot_copy, [23](#)
- disassociate_global_replication_group, [11](#)
- docdb, [4](#)
- domain_metadata, [25](#)
- download_db_log_file_portion, [18](#)
- dynamodb, [5](#)
- dynamodbstreams, [8](#)
- elasticache, [10](#)
- enable_kinesis_streaming_destination, [7](#)
- enable_logging, [23](#)
- enable_snapshot_copy, [23](#)
- execute_sql, [20](#)
- execute_statement, [7](#), [20](#)
- execute_transaction, [7](#)
- export_table_to_point_in_time, [7](#)
- failover_db_cluster, [5](#), [14](#), [18](#)
- failover_global_replication_group, [11](#)
- get_attributes, [25](#)

- get_cluster_credentials, [23](#)
- get_item, [7](#)
- get_records, [9](#)
- get_reserved_node_exchange_offerings, [23](#)
- get_shard_iterator, [9](#)

- import_installation_media, [18](#)
- increase_node_groups_in_global_replication_group, [11](#)
- increase_replica_count, [11](#)
- increase_replication_factor, [3](#)

- list_allowed_node_type_modifications, [11](#)
- list_backups, [7](#)
- list_contributor_insights, [7](#)
- list_domains, [25](#)
- list_exports, [7](#)
- list_global_tables, [7](#)
- list_streams, [9](#)
- list_tables, [7](#)
- list_tags, [3](#)
- list_tags_for_resource, [5, 11, 14, 18](#)
- list_tags_of_resource, [7](#)

- modify_cache_cluster, [11](#)
- modify_cache_parameter_group, [11](#)
- modify_cache_subnet_group, [11](#)
- modify_certificates, [18](#)
- modify_cluster, [23](#)
- modify_cluster_db_revision, [23](#)
- modify_cluster_iam_roles, [23](#)
- modify_cluster_maintenance, [23](#)
- modify_cluster_parameter_group, [23](#)
- modify_cluster_snapshot, [23](#)
- modify_cluster_snapshot_schedule, [23](#)
- modify_cluster_subnet_group, [23](#)
- modify_current_db_cluster_capacity, [18](#)
- modify_db_cluster, [5, 14, 18](#)
- modify_db_cluster_endpoint, [14, 18](#)
- modify_db_cluster_parameter_group, [5, 14, 18](#)
- modify_db_cluster_snapshot_attribute, [5, 14, 18](#)
- modify_db_instance, [5, 14, 18](#)
- modify_db_parameter_group, [14, 18](#)
- modify_db_proxy, [18](#)
- modify_db_proxy_target_group, [18](#)
- modify_db_snapshot, [18](#)
- modify_db_snapshot_attribute, [18](#)
- modify_db_subnet_group, [5, 14, 18](#)
- modify_event_subscription, [14, 18, 23](#)
- modify_global_cluster, [18](#)
- modify_global_replication_group, [12](#)
- modify_option_group, [18](#)
- modify_replication_group, [12](#)
- modify_replication_group_shard_configuration, [12](#)
- modify_scheduled_action, [23](#)
- modify_snapshot_copy_retention_period, [23](#)
- modify_snapshot_schedule, [23](#)
- modify_usage_limit, [23](#)
- modify_user, [12](#)
- modify_user_group, [12](#)

- neptune, [12](#)

- pause_cluster, [23](#)
- promote_read_replica, [18](#)
- promote_read_replica_db_cluster, [14, 18](#)
- purchase_reserved_cache_nodes_offering, [12](#)
- purchase_reserved_db_instances_offering, [18](#)
- purchase_reserved_node_offering, [23](#)
- put_attributes, [25](#)
- put_item, [7](#)

- query, [7](#)

- rds, [15](#)
- rdsdataservice, [19](#)
- rebalance_slots_in_global_replication_group, [12](#)
- reboot_cache_cluster, [12](#)
- reboot_cluster, [23](#)
- reboot_db_instance, [5, 14, 18](#)
- reboot_node, [3](#)
- redshift, [20](#)
- register_db_proxy_targets, [18](#)
- remove_from_global_cluster, [18](#)
- remove_role_from_db_cluster, [14, 18](#)
- remove_role_from_db_instance, [18](#)
- remove_source_identifier_from_subscription, [14, 18](#)
- remove_tags_from_resource, [5, 12, 14, 18](#)

reset_cache_parameter_group, [12](#)
reset_cluster_parameter_group, [23](#)
reset_db_cluster_parameter_group, [5](#), [14](#),
[18](#)
reset_db_parameter_group, [14](#), [18](#)
resize_cluster, [23](#)
restore_db_cluster_from_s3, [18](#)
restore_db_cluster_from_snapshot, [5](#), [14](#),
[18](#)
restore_db_cluster_to_point_in_time, [5](#),
[14](#), [18](#)
restore_db_instance_from_db_snapshot,
[18](#)
restore_db_instance_from_s3, [18](#)
restore_db_instance_to_point_in_time,
[19](#)
restore_from_cluster_snapshot, [23](#)
restore_table_from_backup, [7](#)
restore_table_from_cluster_snapshot,
[23](#)
restore_table_to_point_in_time, [7](#)
resume_cluster, [23](#)
revoke_cache_security_group_ingress,
[12](#)
revoke_cluster_security_group_ingress,
[23](#)
revoke_db_security_group_ingress, [19](#)
revoke_snapshot_access, [23](#)
rollback_transaction, [20](#)
rotate_encryption_key, [23](#)

scan, [7](#)
select, [25](#)
simplifiedb, [24](#)
start_activity_stream, [19](#)
start_db_cluster, [5](#), [14](#), [19](#)
start_db_instance, [19](#)
start_db_instance_automated_backups_replication,
[19](#)
start_export_task, [19](#)
start_migration, [12](#)
stop_activity_stream, [19](#)
stop_db_cluster, [5](#), [14](#), [19](#)
stop_db_instance, [19](#)
stop_db_instance_automated_backups_replication,
[19](#)

tag_resource, [3](#), [7](#)
test_failover, [12](#)
transact_get_items, [7](#)
transact_write_items, [7](#)
untag_resource, [3](#), [7](#)
update_cluster, [3](#)
update_continuous_backups, [7](#)
update_contributor_insights, [7](#)
update_global_table, [7](#)
update_global_table_settings, [7](#)
update_item, [7](#)
update_parameter_group, [3](#)
update_subnet_group, [3](#)
update_table, [7](#)
update_table_replica_auto_scaling, [7](#)
update_time_to_live, [7](#)